A Straw Shows Which Way the Wind Blows: Ranking Potentially Popular Items from Early Votes

Peifeng Yin[†,*] Ping Luo[§] Min Wang[♯] Wang-Chien Lee[‡]

[†,‡] Department of Computer Science & Engineering, Pennsylvania State University
[§,♯] Hewlett Packard Labs China, Beijing, China
{[†] pzy102, [‡] wlee} @cse.psu.edu, {[§] ping.luo, [♯] min.wang6}@hp.com

ABSTRACT

Prediction of popular items in online content sharing systems has recently attracted a lot of attention due to the tremendous need of users and its commercial values. Different from previous works that make prediction by fitting a popularity growth model, we tackle this problem by exploiting the latent *conforming* and *maverick* personalities of those who vote to assess the quality of on-line items. We argue that the former personality prompts a user to cast her vote conforming to the majority of the service community while on the contrary the later personality makes her vote different from the community. We thus propose a *Conformer-Maverick (CM)* model to simulate the voting process and use it to rank top-k potentially popular items based on the early votes they received. Through an extensive experimental evaluation, we validate our ideas and find that our proposed CM model achieves better performance than baseline solutions, especially for smaller k.

Categories and Subject Descriptors

H.4 [**Information Systems**]: Applications; J.4 [**Computer Applications**]: Social and Behavior Sciences

General Terms

Experimentation, Human Factors.

1. INTRODUCTION

The vision of Web 2.0 aims to encourage people to publish and share digital contents in the Internet. Many web and mobile services, e.g., YouTube[1], JokeBox[2], etc, have provided convenient tools to facilitate content sharing. Moreover, they support a voting function to allow users to express their positive/negative opinions, usually in simple binary format such as "like"/"dislike" in YouTube and "Thump up"/"Thump down" in JokeBox, on published items.

Due to the overwhelming quantity and diversity of published items, many of these services tend to put on their service front page the *popular items* that have received the most (positive) votes. By assuming that high-quality items receive more positive than negative votes, this simple strategy addresses the difficulty of assessing the quality of published items. Nevertheless, while putting those items on the front page may facilitate new or inactive users to find popular items easily, it does not help the loyal users who frequently return to the service since they probably have already accessed or even voted those so-called popular items. Moreover, it may lead to the situation of "rich-get-richer" [13], making these popular items difficult to be replaced. Furthermore, "popularity" may imply "obsolete" in some cases. Consider sales on Groupon[3] or Dealsea[4], where super deals usually come with a cap on limited number of participants. There is no doubt that many people will grab these deals, making them more popular and at the same time faster to reach the cap than other items. Business strategies that rely on the "known" popularity of items will unavoidably highlight near-expire deals, which is not desirable. We argue that user experience will be greatly improved if a system can predict the "potential" popularity of an item early without waiting until the popularity of the item becomes known through accumulating many votes. Additionally, being able to predict the popularity of items have tremendous value to not only service providers but also marketers who would bid for ad-space on items with high potential popularity in order to maximize the exposure. In this work, we aim to develop a popularity prediction technique that effectively predicts how likely a newly published item will become popular.

To predict popular items, a content-based technique has been proposed in [18]. In this work, a textual item is split into meaningful words which in turn forms a feature vector of the item. Next, a classification machine is learned to predict whether a new item, given its feature vector, will be popular or not. This method has several deficiencies. Firstly, this technique is not applicable to non-text contents, e.g., audio and video. Secondly, even for textual contents, it may not handle short messages such as tweets and short jokes where discriminative feature vectors are difficult to form due to the limited number of words or the subtle implications. Finally, the content-based method is unable to process items with new words that have not appeared in existing items.

Another idea for popularity prediction is to exploit the early votes. In [13], the trend of the vote growth is explored

*The work was done when Peifeng Yin was visiting HP Labs China.

[1] http://www.youtube.com

[2] http://itunes.apple.com/us/app/all-in-1-joke-box-no-ads-unlimited/id363494433?mt=8

[3] http://www.groupon.com

[4] http://dealsea.com

WSDM'12, February 8–12, 2012, Seattle, Washington, USA.

and a logarithm linear model is learned to fit the correlation between early votes and the popularity. Thus, popularity prediction can be made by this model. However, designed for predicting the growth of item groups, this method is prune to possibly high inaccuracy when targeting on the individual item. Our approach, also exploiting the early votes, aims to rank the potential of individual item to become popular, instead of focusing on the popularity growth trend.

Through an empirical study we observe that, while there does exist correlation between early and final vote status, the people who contribute the early votes can be exploited to improve the prediction accuracy. Specifically, we observe that among early voters, some people's votes tend to be conforming to the opinions of the majority in the user community while some others exhibit contrary voting behavior. Thus, in this paper, we refer to the users exhibiting high degree of conforming behavior as *Conformer* and name those who tend to cast votes differently from the opinions of the society or social group (the user community of a service in our study) as *Maverick*. As shown in Figure 1, for Conformers, since their behaviors tend to be conforming to the group behavior, their votes usually can be amplified to represent the group opinions. On the other hand, the Mavericks give votes opposite to the majority, i.e., they favor items which do not attract the majority of users while giving negative ratings to items which are actually popular among the majority. For example, in the case of JokeBox where users share and rate funny jokes, Conformers are amused by funny jokes while the Mavericks may laugh at some clueless ones or shaggy-dog stories.

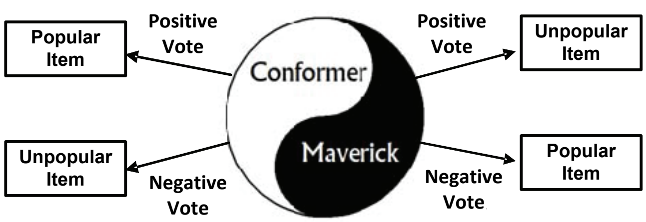


Figure 1: Illustration of Conformer and Maverick

We argue that each person has both conforming and maverick personalities in different degrees and thus propose a *Conformer-Maverick (CM)* model to capture the two personalities of a person. In this model, we assume that each person is equipped with these two personalities and one of them prevails when casting a vote. As mentioned, different people have these two personalities in varied degrees, which are learned by maximizing the probability of the observed vote data with EM algorithm. To predict whether an item will be popular or not, the conformer-maverick preference of its early voters are checked. Ideally, if positive votes come from more conformers and/or negative votes from more mavericks, the item is more likely to be welcomed by the majority, and vice versa.

Another challenge for popular-item prediction is how the popularity is measured based on the aggregated votes. In previous works [13, 18, 9, 14] only positive votes are available for an item and thus this number is naturally treated as its popularity. In this study, however, an item can receive both positive and negative votes and it remains a problem how such two-dimensional values are converted to a single dimensional popularity value. We designed a popularity measurement (detailed in Section 2) that considers both positive and negative votes and represent an item's popularity with a real number ranging from -1 to 1.

To sum up, our contribution is listed below:

- We designed a new popularity measure of items by combining both of their positive and negative votes.
- We proposed a Conformer-Maverick model to capture two personalities of a user in the voting process.
- With the CM model, we developed two ranking mechanisms, i.e., Aggregation-based ranking and Q-based ranking. The first one predicts the vote for each user and aggregates them while the second one directly predicts the item's popularity degree defined in Section 2.
- We conducted extensive experiments to evaluate the performance of the proposed solutions.

The rest of the paper is organized as follows. Section 2 defines the popularity measurement equation and formulates the problem, i.e., top-k popular item ranking. Section 4 and 5 respectively discuss the details of the proposed CM model and how it is adopted to accomplish the popularity prediction. Section 3 gives details of Naive CM model. Evaluation is put in Section 6. Section 7 gives related work and finally Section 8 concludes the paper and introduces our future work.

2. PRELIMINARIES

In this section we formally define a measure of item popularity and then formulate the research problem of *top-k popular item prediction*. Finally we analyze the problem and motivate our ideas behind the proposed solutions.

2.1 Popularity Measurement

In a content sharing and rating system, the popularity of a published item may be reflected by the votes it receives. Intuitively, the more positive votes it receives, the more popular it is and vice versa. While one may argue that an item receiving a large number of votes (whether they are positive or not) is popular, in this paper, we define popular items as those that are "liked" by the majority of the service users, i.e., those that receive many more positive than negative votes. To judge whether an item is liked by the majority, one precondition is that the item has received sufficient number of votes. For example, it is not reasonable to refer to an item receiving only 2 positive but 0 negative votes as popular. In other words, our measure of popularity aim to ensure that the number of votes exceeds certain threshold.

Formally, in a content sharing system, given pv positive votes and nv negative votes of an item where $pv + nv \geq \sigma$, a vote threshold, its *popularity* q is computed as in Equation (1).

$$q = \begin{cases} \frac{pv-nv}{pv+nv+1}; & pv - nv \neq 0 \\ \frac{\epsilon}{pv+nv+1}; & pv - nv = 0 \end{cases} \tag{1}$$

where ϵ is a small constant, e.g., 0.01. Note that this measure employs a threshold σ to govern the minimum total number of votes, which are system/service-dependent. We discuss how to determine the setting of σ in Section 6.

As shown Equation (1), the popularity ranges from -1 to 1. A measured popularity close to 1 indicates significant number of positive votes over the negative ones, and vice versa. On the other hand, a popularity value close to 0 suggests that the item is rather controversial, i.e., the number of positive and negative votes are quite even, and thus difficult to classify whether it is popular or not. Notice that, in Equation (1), we introduce a constant ϵ when an item receives equal number of positive and negative votes, aiming to incorporate the *controversial degree*, which is inversely proportional to the number of total votes, into the measure of item popularity.

We use an example in Table 1 to illustrate the popularity measures for different items. Suppose that there are six items and the votes they received are listed in the table. We can see that the popularity of items (i_1, i_2, i_3) are positive while that of items (i_4, i_5, i_6) are negative, which follows the common wisdom that more positive (negative) votes indicates higher (lower) popularity. Moreover, the items with popularity very close to 0 is more controversial than those with popularity less close to 0, e.g., item i_4 is more controversial than i_3.

Table 1: Illustration of Item Popularity

Item	Positive Vote	Negative Vote	Popularity
i_1	14	1	0.8125
i_2	97	2	0.95
i_3	4	3	0.125
i_4	99	100	-0.005
i_5	5	10	-0.3125
i_6	2	97	-0.95

2.2 Problem Formulation

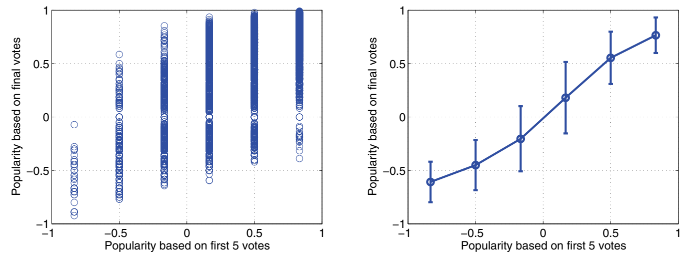
Here we formulate the research problem of predicting the most popular items among newly created ones as a ranking problem. Our goal is to design a ranking function that quantifies the potential of a newly published item to become popular based on its early votes. Particularly, given N newly published items associated with their early n votes, the algorithm returns k items that are most likely to be popular.

Top-k Popular Item Ranking. Let the item i's first n votes be denoted by $\vec{V_i} = \langle (u_{i_1}, v_{i_1}), \cdots, (u_{i_n}, v_{i_n}) \rangle$, where $u_{i_j}, 1 \le j \le n$ is a voter and $v_{i_j} \in \{-1, 1\}$ is the vote casted by u_{i_j}. Given a set of items $S = \{\vec{V_1}, \cdots, \vec{V_N}\}$, the Top-k Popular Item Ranking determines a sorted list of k items $S' \subseteq S, |S'| = k$ such that i) any item in S' is more likely to be popular than all the items in $S - S'$ ii) items in S' are non-increasingly sorted by their potential to be popular.

2.3 Problem Analysis

To predict a candidate item's potential to become popular based on its early votes, a natural question is whether there is a correlation between the early and final votes. To answer this question, we collect voting data from a popular iPhone application JokeBox[5] to conduct an analysis on the items which have received no less than 8 votes. We plot their early and final votes on a graph where x and y axis respectively represent the popularity computed according to early and final votes (as defined in Equation (1)). The results are shown in Figure 2, where Figure 2(a) displays the raw plot and Figure 2(b) shows the mean value and standard deviation. Generally speaking, items with good early votes may likely turn out to be popular. However, the scattered distribution of the data points in Figure 2(a) suggests potential inaccuracy if prediction is purely based on the votes. For example, given two newly published items, one with 3 positive and 2 negative early votes (i.e., $x \approx 0.1667$) while the other one's with 4 positive and 1 negative early votes (i.e., $x = 0.5$). It is however difficult to judge which one is more likely to be popular since statistically either one can turn out to be more popular than the other. Furthermore, if two items' early votes are exactly the same, it is unable to distinguish them.

[5]Details of the data are introduced in Section 6



(a) Raw Data (b) Statistical Data

Figure 2: Statistical analysis on correlation where x and y axis respectively represent the popularity computed according to early and final votes.

In this study we resort to the voters to refine the prediction, aiming to find users who always give "conforming" (or "maverick") votes to items. Here the conforming votes mean that the casted positive/negative early votes later turn out to receive more positive/negative votes eventually. On the contrary, maverick votes mean that the casted positive/negative early votes leads to more negative/positive votes eventually. Thus, we aim to identify users whose votes are either positively or negatively correlated to the popularity of items. To this end, we calculate the *Pearson correlation coefficient*[6] of users' votes and items' popularity. Formally, given the votes of a user on multiple items (denoted by $\mathbf{X}$) and the corresponding items' popularity (denoted by $\mathbf{Q}$), we use Pearson correlation coefficient to measure the strength of linear dependence between $\mathbf{X}$ and $\mathbf{Q}$. The absolute value of Pearson correlation coefficients is no bigger than 1. Values equal to 1 or -1 correspond to purely linear positive and negative correlations respectively.

Using the voting data collected from JokeBox, we calculate the Pearson correlation coefficient for users randomly sampled from the whole user set. The distribution of this correlation values is plotted in Figure 3. We can see that most users have positive correlation values around 0.5. This indicates that people are not purely a conformer or a maverick but a combination of the two biased towards the conformer personality. Based on this observation, we propose a *Conformer-Maverick (CM)* model to capture these two personalities and adopt the model in popularity prediction.

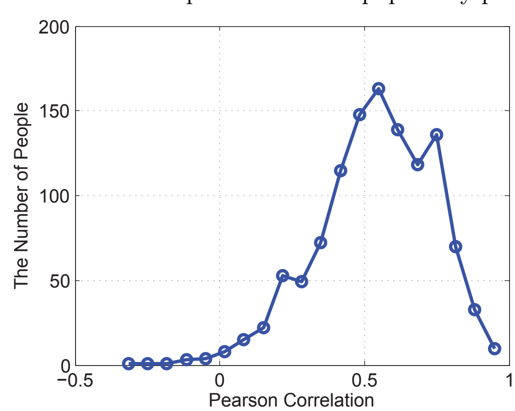


Figure 3: Pearson correlation of user's votes and items' quality

3. NAIVE CONFORMER-MAVERICK MODEL

In this section, we introduce a solution to the top-k popular item prediction problem. Based on our observations

[6]http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient

discussed in Section 2.3, here we first propose a heuristic method to model the two user personalities of conformers and mavericks and develop a ranking function for popular item prediction. We refer to the method as *Naive Conformer-Maverick (NCM)* model to differentiate it from another solution discussed later in Section 4.

Recall that when a Conformer (Maverick) gives positive (negative) vote to an item, it turns out to be popular later, and vice versa. However, most people are not a pure Conformer or Maverick but their combinations (as shown in Figure 3). A question is how much we could rely on a person's view when she gives a positive (negative) vote to an item.

The NCM assumes that a person has a Conformer weight $w_c \in [0,1]$ and a Maverick weight $w_m \in [0,1]$. The difference of the two weights suggests how reliable the person's opinion is. A large value of $w_c - w_m$ indicates that the person's vote v reflects the popularity q of the item, i.e., $v = 1$ indicates a positive q while $v = -1$ a negative one. Formally, given a series of the votes $\mathbf{X} = \{x_1, \cdots, x_n\}$ casted by a user and the popularity of the voted items $\mathbf{Q} = \{q_1, \cdots, q_n\}$, the w_c and w_m of the user is defined in Equation (2).

$$w_c = \frac{C + \sum_{i=1}^{n} \mathbf{1}_{x_i \cdot q_i > 0} \cdot |q_i|}{2C + n}, w_m = \frac{C + \sum_{j=1}^{n} \mathbf{1}_{x_j \cdot q_j < 0} \cdot |q_i|}{2C + n} \tag{2}$$

where C is a constant larger than 1, serving as a prior weight and $\mathbf{1}_{\text{condition}} = \begin{cases} 1, & \text{condition is true} \\ 0, & \text{otherwise} \end{cases}$

Here we do not calculate w_c and w_m via the Pearson correlation of $\mathbf{X}$ and $\mathbf{Q}$ because the calculation of this correlation may fail when entries in $\mathbf{X}$ are all equal to a constant value (1 or -1) due to the issue of zero denominator.

For a new user who has not committed any single vote, her Conformer and Maverick weights are initialized as $w_c = w_m = \frac{C}{2C} = 0.5$, indicating an unclear personality. When she starts to vote on items, her weights will gradually change. Specifically, each time when her vote x is consistent with an item's final vote q, i.e., $x \cdot q > 0$, her Conformer weight w_c will increase. Otherwise, her Maverick weight w_m will increase as $x \cdot q < 0$.

As for the top-k popularity prediction, given an item's first n early votes $\vec{V_i} = \langle (u_{i_1}, v_{i_1}), \cdots, (u_{i_n}, v_{i_n}) \rangle$, where $v_{i_j} \in \{-1, 1\}$ is a vote and u_{i_j} is the voter with Conformer and Maverick weights as $w_c^{i_j}$ and $w_m^{i_j}$, respectively, the popularity ranking score is the weighted sum of those votes as defined in Equation (3).

$$\texttt{PopRank}(\vec{V_i}) = \sum_{j=1}^{n} (w_c^{i_j} - w_m^{i_j}) \cdot v_{i_j} \tag{3}$$

We use a simple example to illustrate the ranking function. Suppose there are two new items whose early votes are respectively $(u_1, 1)$ and $(u_2, 1)$, where the first voter's Conformer and Maverick weights are $w_c^1 = 0.9$ and $w_m^1 = 0.1$, while the second one's are $w_c^2 = 0.2$ and $w_m^2 = 0.8$. The ranking of the two items are shown below.

$$\begin{aligned} \texttt{PopRank}(\vec{V_1}) &= (0.9 - 0.1) \times 1 = 0.8 \\ \texttt{PopRank}(\vec{V_2}) &= (0.2 - 0.8) \times 1 = -0.6 \end{aligned} \tag{4}$$

Therefore the first item is more likely to be popular.

4. CONFORMER-MAVERICK MODEL

In this section we propose a probabilistic model, *Conformer-Maverick Model*, to simulate the voting process. Different from the NCM model that adopts a heuristic way to model the user profile and rank the item, this new model considers the voting behavior as a constrained random process. Based on the model, we aim to learn a better assignment of the conformer and maverick personalities for each user.

In an item voting process, a user firstly views an item, forms a judgement according to its quality, and casts a vote based on the judgement. For conformer-minded users, their votes usually are aligned with the general opinions of the user community while for maverick-dominated users, their votes are rather deviated. This explains why people with different personalities cast different votes on the same item, as shown in Figure 1. To model the voting behavior as a constrained random process, jointly determined by the voter's personality and the item's quality, we assume that one person's voting behavior is independent of others'. Figure 4 illustrates the process, where u stands for a user and z is her latent personality (i.e., conformer or maverick), which, together with the item's quality q, generates a parameter p. This p finally determines whether the vote v is positive or negative.

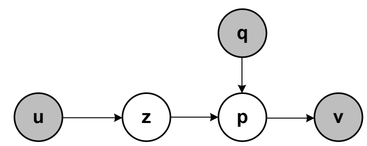


Figure 4: Generative Model for Voting

Specifically, the generating process is shown in Algorithm 1. To cast a vote in one trial, the voter first selects a latent personality from Conformer z_c and Maverick z_m in accordance with a multinomial distribution $\texttt{Multinomial}(\pi_c, \pi_m)$. Then, given the item quality q and the selected topic z, a parameter p, ranging from 0 to 1, is randomly generated from a generation function $\texttt{G}_p(z, q)$ (detailed later), where p indicates the probability that the person may give a positive vote. Finally, a vote is generated by a Bernoulli process.

Particularly, the key to the above generative process is the p-generation function $\texttt{G}_p(z, q)$, which should satisfy the following requirements. Firstly, the ranges of its output should be between 0 and 1 as the p represents the probability of casting a positive vote. Secondly, when the Conformer personality z_c is selected, the $p = \texttt{G}_p(z_c, q)$ should be positively proportional to q. Meanwhile, when the Maverick personality z_m is selected, p should be negatively correlated with q.

Algorithm 1 Process of vote generation

1. Choose a personality $z \sim \texttt{Multinomial}(\pi_c, \pi_m)$, $z \in \{z_c, z_m\}$
2. Choose a parameter $p \sim \texttt{G}_p(z, q)$, $p \in [0, 1]$
3. Choose a vote $v \sim \texttt{Bernoulli}(p)$, $v \in \{-1, 1\}$

In the rest of the section, we first introduce the modeling of latent topics, then describe the process of generating p with the parameters z and q, i.e., the function $\texttt{G}_p(z, q)$. Finally we discuss the methods of parameter learning.

4.1 Modeling the Latent Personalities

There are two latent personalities in the space for each person, corresponding to two inclinations of a person when she makes decisions. Specifically they are referred to as *Conformer* $z_1 = 1$ and *Maverick* $z_2 = -1$. The Conformer personality dictates that the person is more likely to conform with the majority when expressing her opinions, e.g., voting an item. On the other hand, the Maverick personality makes the person behave differently from most people. For instance, in a joke-sharing system where popular items are funny jokes, a person may give positive vote to jokes that are liked by many others and give negative vote to cold ones when Conformer is "active".On the contrary, a person may thumb up the cold jokes and thumb down funny jokes when her Maverick personality takes in charge.

We assume that all people have these two personalities and different people may have different probability distributions on Conformer and Maverick due their personalities. Two parameters $\pi_1, \pi_2 \in [0,1]$ are used to respectively represent the probability that the person chooses z_1 or z_2 when she votes an item. It is easily seen that $\pi_1 + \pi_2 = 1$.

4.2 Modeling the p-Generation Process

As mentioned above, given a selected personality z and an item whose quality is q, a variable $p \in [0,1]$ will be generated. We assume this generation process satisfies a Beta distribution shown in Equation (5).

$$\mathtt{Be}(p;a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1-p)^{b-1} = \frac{p^{a-1}(1-p)^{b-1}}{\mathtt{B}(a,b)} \quad (5)$$

where a and b are linearly associated with z and q.

In probability theory, Beta distribution is often used to describe a prior distribution of a parameter for some distribution, e.g., Bernoulli distribution. Specifically, the a and b jointly determine the probability distribution of p. When $a > b \geq 1$, the value of p is more likely to be large. In case of $1 \leq a < b$, the value of p is closer to 0.

Given z and q, the values of a and b are computed according to Equation (6).

$$a = f_a(z,q) = C \times H(z \cdot q), \; b = C - a \quad (6)$$

where C is a predefined constant bigger than 1 and function $H(x)$ is a normalized function defined in Equation (7).

$$H(z \cdot q) = \frac{z \cdot q - (-1)}{1 - (-1)} = \frac{z \cdot q + 1}{2} \quad (7)$$

It normalizes the original range $[-1,1]$ of $z \cdot q$ into $[0,1]$.

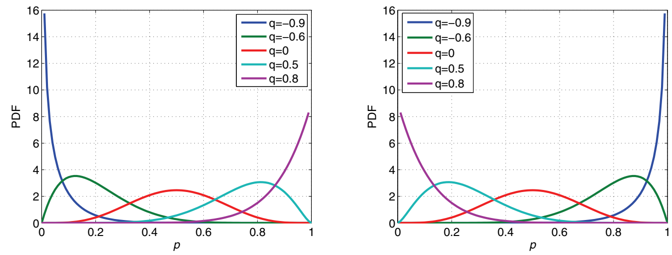
To sum up, the generation function is defined below.

$$\mathtt{G}_p(z,q) = \frac{\Gamma(C)}{\Gamma(\frac{q \cdot z+1}{2} \cdot C)\Gamma(\frac{-q \cdot z+1}{2} \cdot C)} p^{\frac{q \cdot z+1}{2} \cdot C-1}(1-p)^{\frac{-q \cdot z+1}{2}C-1} \quad (8)$$

As seen in Equations (6) and (7), a and b are respectively positively and negatively correlated with the value of $z \cdot q$. When the personality of Conformer z_c is selected, a large q indicates a large a and a small b, which thus leads to a high probability of positive vote. On the contrary, when Maverick z_m is selected, high quality results in small a and large b. Thus, a negative vote is more likely to happen. The probability density function for p with regarding to different q and z is shown in Figure 5 where C is set to 10. As you can see, the density functions coincide with our intuitive understandings on $\mathtt{G}_p(z,q)$. It is worth mentioning that, as to be shown next, all the derivations on this model is irrelevant to the parameter C.

4.3 Modeling the Voting Process

With a generated probability p, the voting is modeled as a Bernoulli process, i.e., $Pr(x=1) = p$ and $Pr(x=-1) = 1-p$. Aggregating all the processes, i.e., personality selection, p-generation and voting, we can obtain the complete



(a) $z = z_c$ (b) $z = z_m$

Figure 5: Illustration of $\mathtt{G}_p(z,q)$

probability that the person votes 1 in Equation (9).

$$\begin{aligned} Pr(x=1,z|q) &= \sum_{j=1}^{2} I_{z=z_j} Pr(x=1,z|q) \\ &= \sum_{j=1}^{2} I_{z=z_j} \left(\int_0^1 Pr(x=1|p)Pr(p|z,q)dp \right) \pi_j \\ &= \sum_{j=1}^{2} I_{z=z_j} \left(\int_0^1 p \cdot \mathtt{Be}(p; f_a(z,q), f_b(z,q))dp \right) \pi_j \end{aligned} \quad (9)$$

where $I_{z=z_j} = \begin{cases} 1 & z = z_j \\ 0 & \text{otherwise} \end{cases}$

Note that the expected value of Beta distribution $\mathtt{Be}(p;a,b)$ is $\frac{a}{a+b}$, i.e., $\int_0^1 p \cdot \mathtt{Be}(p;a,b)dp = \frac{a}{a+b}$. Equation (9) can thus be reduced to Equation (10).

$$\begin{aligned} Pr(x=1,z|q) &= \prod_{j=1}^{2} \left(\frac{f_a(z,q)\pi_j}{f_a(z,q)+f_b(z,q)} \right)^{I_{z=z_j}} \\ &= \prod_{j=1}^{2} \left(\frac{\pi_j(z \cdot q+1)/2}{(z \cdot q + 1 - z \cdot q + 1)/2} \right)^{I_{z=z_j}} = \prod_{j=1}^{2} (H(z \cdot q) \cdot \pi_j)^{I_{z=z_j}} \end{aligned} \quad (10)$$

Similarly, the probability that the person gives a negative vote is shown in Equation (11).

$$\begin{aligned} Pr(x=-1,z|q) &= \sum_{j=1}^{2} I_{z=z_j} \left(\int_0^1 Pr(x=-1|p)Pr(p|z,q)dp \right) \pi_j \\ &= \sum_{j=1}^{2} I_{z=z_j} \left(\int_0^1 (1-p) \cdot \mathtt{Be}(p; f_a(z,q), f_b(z,q))dp \right) \pi_j \\ &= \prod_{j=1}^{2} \left(\frac{f_b(z,q)\pi_j}{f_a(z,q)+f_b(z,q)} \right)^{I_{z=z_j}} = \prod_{j=1}^{2} ((1-H(z \cdot q)) \cdot \pi_j)^{I_{z=z_j}} \end{aligned} \quad (11)$$

Note that $H(x) + H(-x) = 1$ and Equation (10) and (11) can thus be unified as in Equation (12).

$$Pr(x,z|q) = \prod_{j=1}^{2} (H(x \cdot z \cdot q) \cdot \pi_j)^{I_{z=z_j}} \quad (12)$$

Suppose that, in a rating system, a user rates a collection of items whose qualities are $\mathbf{Q} = \{q_1, \cdots, q_n\}$. Let $\mathbf{X} = \{x_1, \cdots, x_n\}$ denote the observed votes the user gives and $\mathbf{Z} = \{z_1, \cdots, z_n\}$ denote the particular topic that dominates the i_{th} voting. Note that $x_i, 1 \leq i \leq n$ are binary values, i.e., $x_i \in \{1, -1\}$.

Thus the probability of these votes can be computed as in Equation (13).

$$Pr(\mathbf{X},\mathbf{Z}|\mathbf{Q}) = \prod_{i=1}^{n} Pr(x_i, z_i|q_i) = \prod_{i=1}^{n}\prod_{j=1}^{2}(H(x_i \cdot z_i \cdot q_i)\cdot \pi_j)^{\tau_{ij}} \tag{13}$$

where the indicator $\tau_{ij}(1 \le i \le n, j \in \{1,2\})$ is defined as

$$\tau_{ij} = \begin{cases} 1 & \text{the } i^{th} \text{ vote is generated by } j^{th} \text{ topic} \\ 0 & \text{otherwise} \end{cases}$$

4.4 Learning the Model Parameters

We use maximum-likelihood to find the model parameters, i.e., finding proper parameter values to maximize the probability of observed data. The objective function is defined in Equation (14).

$$\begin{aligned} &\mathfrak{L}(\pi_1,\pi_2;\mathbf{X},\mathbf{Z},\mathbf{Q}) = \log Pr(\mathbf{X},\mathbf{Z}|\mathbf{Q};\pi_1,\pi_2) + \lambda(1-\pi_1-\pi_2) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{2}\tau_{ij}\{\log(H(x_i \cdot z_i \cdot q_i)) + \log \pi_j\} + \lambda(1-\pi_1-\pi_2) \end{aligned} \tag{14}$$

where λ is a Lagrange multiplier.

We use Expectation-maximization (EM) algorithm [4] to solve the above equation. Let the parameters at t^{th} iteration be denoted by $\theta^{(t)}$. The two steps are shown below.

E-step

$$\begin{aligned} E(\tau_{ij}^{(t+1)}) &= Pr(z_j|x_i,q_i) = \frac{Pr(x_i,z_j|q_i)}{Pr(x_i,z_1|q_i)+Pr(x_i,z_2|q_i)} \\ &= \frac{H(x_i \cdot z_j \cdot q_i)\cdot \pi_j^{(t)}}{\sum_{k=1}^{2} H(x_i \cdot z_k \cdot q)\cdot \pi_k^{(t)}} \end{aligned} \tag{15}$$

M-step

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} E(\tau_{ij})}{\sum_{i=1}^{n} E(\tau_{i1}) + \sum_{i=1}^{n} E(\tau_{i2})} = \frac{\sum_{i=1}^{n} E(\tau_{ij})}{n} \tag{16}$$

In real situation, however, the quality of an item is usually impossible to measure. In this work we use the popularity degree defined in Equation (1) to represent the quality by assuming that a high-quality item often receives more positive and less negative votes.

Another issue worth attention is the difference between NCM and CM in modeling the user profiles. For NCM, each vote is either contributed to Conformer weight w_c or Maverick weight w_m, as shown in Equation (2). It is a coarse assignment method of weights. Compared to this "radical attitude" behind NCM, in CM each observed vote contributes probabilistically to both π_c and π_m as shown in Equations (15) and (16). Specifically, for a vote x_i on an item with the popularity q_i, the value of $Pr(z_c|x_i,q_i)$ contributes its conformer personality while that of $Pr(z_m|x_i,q_i)$ contributes its maverick personality. As such, CM provides a fine-grained assignment method of weights. It preserves the probability that a person can give "Conformer/Maverick-minded" vote even if the opposite personality is selected. This robustness leads to CM's better performance to NCM, as to be shown later in Section 6.4.

4.5 Discussion

In this section we provide a detailed discussion on the proposed model. We assume that the item quality satisfies a continuous uniform distribution over the interval $[-1,1]$.

The CM model discussed above assumes that each person has two topics, i.e., Conformer and Maverick. In other words, the user profile is modeled as a suit of two topics together with the topic distribution. Formally, the i^{th} user profile is denoted as $u_i = \langle \pi_1^i, \pi_2^i \rangle$, where $\pi_j^i, j \in \{1,2\}$ stands for $Pr(z_j|u_i)$, the probability of selecting topic z_j for the i^{th} user.

With this model, we are particularly interested in the following two questions:

- Given an item j with *unknown* quality q_j, what is the probability for the person u_i to give a positive vote?
- Given an item j with *known* quality q_j, what is the probability for the person u_i to give a positive vote?

Let $v_{ij} \in \{-1,1\}$ denote the vote user u_i gives to item j. For the first question, we can compute the probability of positive vote as in Equation (17).

$$\begin{aligned} Pr(v_{ij}=1|u_i) &= \int_{-1}^{1}\sum_{k=1}^{2} Pr(v_{ij}=1|z_k,q)Pr(z_k|u_i)Pr(q)dq \\ &= \frac{1}{2}\int_{-1}^{1}\sum_{k=1}^{2} H(z_k \cdot q)\pi_k^i dq = \frac{1}{2}\sum_{k=1}^{2}\pi_k^i \int_{-1}^{1}\frac{z_k \cdot q + 1}{2}dq \\ &= \frac{1}{2}\sum_{k=1}^{2}\pi_k^i\left(\frac{(z_k+1)^2-(-z_k+1)^2}{4z_k}\right) = \frac{1}{2} \end{aligned} \tag{17}$$

Similarly, we can obtain that

$$Pr(v_{ij}=-1|u_i) = 1 - Pr(v_{ij}=1|u_i) = \frac{1}{2} \tag{18}$$

This result is reasonable because for a sensible person, her probability of liking or disliking an item should be equal as the quality of the item is unknown.

For a given item with known quality q_j, the probability of positive vote is computed in Equation (19).

$$\begin{aligned} Pr(v_{ij}=1|u_i,q_j) &= \sum_{k=1}^{2} Pr(v_{ij}=1|z_k,q_j)Pr(z_k|u_i) \\ &= \sum_{k=1}^{2}\int_{0}^{1} Pr(v_{ij}=1,p|z_k,q_j)dp \cdot Pr(z_k|u_i) = \sum_{i=1}^{2} H(z_i \cdot q_j)\pi_k^i \\ &= \sum_{k=1}^{2}\frac{z_k \cdot q_j + 1}{2}\pi_k^i = \frac{(\pi_1^i - \pi_2^i)\cdot q_j + 1}{2} \end{aligned} \tag{19}$$

Similarly the probability of negative vote is shown below.

$$Pr(v_{ij}=-1|u_i,q_j) = \frac{(\pi_2^i - \pi_1^i)\cdot q_j + 1}{2} \tag{20}$$

Suppose $\pi_1^i > \pi_2^i$, i.e., the person u_i is more of a conformer, the probability of positive (negative) vote increases (decreases) with the growth of the given item's quality q_j. On the other hand, if $\pi_1^i < \pi_2^i$, the person's vote is usually conflicted with the item's quality, suggesting her maverick personality. Finally, if $\pi_1^i = \pi_2^i$, the voting is purely random. Therefore, to help predict the popularity of the item, the favorite voters are those whose π_1^i and π_2^i differ a lot.

5. RANKING POTENTIAL POPULARITY

The key point behind our popularity prediction idea is to design a ranking function that can rank each candidate item's potential popularity based on its early votes. In this section, we give two ranking functions based on the proposed CM model introduced in Section 4.

The first one, referred to as *Aggregation-based Ranking Function*, aggregates the vote values over all the users given the early votes on item i (denoted by $\vec{V_i}$) and voters (denoted by U), i.e., $\sum_{j=1}^{|U|} E(v_{i_j}|u_{i_j}, \vec{V_i})$, where $E(v_{i_j}|u_{i_j}, \vec{V_i})$ is the expected value of user u_{i_j}'s vote on item i given $\vec{V_i}$. The second one, referred to as *Q-based Ranking Function*, defines an item's ranking score as $Pr(q_i \geq q^*|\vec{V_i})$, i.e., the probability that this item's popularity is no smaller than a predefined threshold q^* given the early votes and voters.

As seen from the definitions of these two ranking functions, we consider the early voting values and the personalities of the voters in this prediction. External factors such as social influence [10] and voting time [13] are out of scope of this work. Also, our prediction is focused on the newly published items which are typically listed reverse-chronologically in the "latest stories" list and and thus viewed by users with the same probability.[7]

5.1 Aggregation-based Ranking

To predict one item's potential popularity, an intuitive way is to predict the individual vote and aggregate all of them. Specifically, given an item i's early votes $\vec{V_i}$, we would like to compute the expected vote value for a user u_{i_0}.

$$\begin{aligned} &E(v_{i_0}|u_{i_0}, \vec{V_i}) = Pr(v_{i_0} = 1|u_{i_0}, \vec{V_i}) - Pr(v_{i_0} = -1|u_{i_0}, \vec{V_i}) \\ &= 2Pr(1|u_{i_0}, \vec{V_i}) - 1 = 2\frac{\int_{-1}^{1} \prod_{j=0}^{n} Pr(v_{i_j}|u_{i_j}, q_i)dq_i}{\int_{-1}^{1} \prod_{j'=1}^{n} Pr(v_{i_{j'}}|u_{i_{j'}, q_i})dq_i} - 1 \end{aligned} \tag{21}$$

Now the key is to compute $\prod_{j=1}^{n} Pr(v_{i_j}|q_i, u_{i_j})$, which is shown in Equation (22).

$$\begin{aligned} &\prod_{j=1}^{n} Pr(v_{i_j}|q_i, u_{i_j}) = \prod_{j=1}^{n}\sum_{k=1}^{2} Pr(v_{i_j}|z_k, q_i)Pr(z_k|u_{i_j}) \\ &= \prod_{j=1}^{n}\sum_{k=1}^{2} H(v_{i_j}\cdot z_k \cdot q_i)\pi_k^{i_j} = \sum_{k_1\cdots k_n=1}^{2}\prod_{j=1}^{n} H(v_{i_j}\cdot z_{k_j}\cdot q_i)\cdot \pi_{k_j}^{i_j} \\ &= \sum_{k_1\cdots k_n=1}^{2} (H(q_i))^{\sum_{j=1}^{n}\mathbf{1}_{v_{i_j} z_{k_j}=1}} (H(-q_i))^{\sum_{j=1}^{n}\mathbf{1}_{v_{i_j} z_{k_j}=-1}} \prod_{j=1}^{n}\pi_{k_j}^{i_j} \\ &= \sum_{k_1\cdots k_n=1}^{2} (H(q_i))^{a_{\mathbf{k}}-1}(1-H(q_i))^{b_{\mathbf{k}}-1}\prod_{j=1}^{n}\pi_{k_i}^{i_j} \end{aligned} \tag{22}$$

where the function $a_{\mathbf{k}} - 1 = \sum_{j=1}^{n}\mathbf{1}_{v_{i_j} z_{k_j}=1}$ counts the number of positive ones when multiplying v_{i_j} by z_{k_j} while $b_{\mathbf{k}} - 1$ counts the number of negative ones.

The integration is computed as Equation (23).

$$\begin{aligned} &\int_{-1}^{1}\prod_{j=1}^{n} Pr(v_{i_j}|q_i, u_{i_j})dq_i \\ &= \sum_{k_1\cdots k_n=1}^{2} 2\prod_{j=1}^{n}\pi_{k_j}^{i_j}\int_{H(-1)}^{H(1)} t^{a_{\mathbf{k}}-1}(1-t)^{b_{\mathbf{k}}-1}dt \\ &= 2\sum_{k_1\cdots k_n=1}^{2}\prod_{j=1}^{n}\pi_{k_j}^{i_j}(\mathtt{B}(a_{\mathbf{k}}, b_{\mathbf{k}}) - \mathtt{B}_{H(-1)}(a_{\mathbf{k}}, b_{\mathbf{k}})) \end{aligned} \tag{23}$$

where $\mathtt{B}(a,b)$ and $\mathtt{B}_x(a,b)$ are respectively the complete and incomplete beta function defined in Equation (24).

$$\begin{aligned} \mathtt{B}(a,b) &= \int_0^1 t^{a-1}(1-t)^{b-1}dt = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \\ \mathtt{B}_x(a,b) &= \int_0^x t^{a-1}(1-t)^{b-1}dt \end{aligned} \tag{24}$$

Particularly, when a and b are both integers, $\mathtt{B}_x(a,b)$ can be computed as below:

$$\begin{aligned} &\mathtt{B}_x(a,b) = \mathtt{B}(a,b)\cdot\frac{\mathtt{B}_x(a,b)}{\mathtt{B}(a,b)} = \mathtt{B}(a,b)\cdot\mathtt{I}_x(a,b) \\ &= \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}(\sum_{j=a}^{a+b-1}\frac{(a+b-1)!}{j!(a+b-1-j)!}x^j(1-x)^{a+b-1-j}) \end{aligned} \tag{25}$$

Therefore, the ranking function in Equation (21) can be written as below.

$$E(v_{i_0}|u_{i_0}, \vec{V_i}) = 2\frac{\sum_{k_0\cdots k_n=1}^{2}\prod_{j=0}^{n}\pi_{k_j}^{i_j}\mathtt{B}(a_{\mathbf{k}}, b_{\mathbf{k}})}{\sum_{k_1'\cdots k_n'=1}^{2}\prod_{j'=1}^{n}\pi_{k_{j'}'}^{i_{j'}}\mathtt{B}(a_{\mathbf{k'}}, b_{\mathbf{k'}})} - 1 \tag{26}$$

The ranking score of the item is the aggregation of all individuals' expected votes, as shown in Equation (27).

$$\mathtt{PopRank}(\vec{V_i}) = \sum_{j=1}^{|U|} E(v_{i_j}|u_{i_j}, \vec{V_i}) \tag{27}$$

where $|U|$ is the total number of users in the system.

5.2 Q-based Ranking

An alternative way of popularity prediction, referred to as *Q-based ranking*, is to estimate a particular item's popularity degree q defined in Equation (1). For a predefined popularity threshold q^*, we want to calculate the probability that the particular item's popularity is larger than q^*.

Let $\vec{V_i} = \langle(u_{i_1}, v_{i_1}), \cdots, (u_{i_n}, v_{i_n})\rangle$ denote the first n votes for item i where $v_{i_j} \in \{1, -1\}$ is given by user u_{i_j}, whose personality distribution parameters are represented as $\langle\pi_{i_j}^1, \pi_{i_j}^2\rangle$. We also assume that the popularity of the item satisfies a uniform distribution between -1 to 1. The probability can be written as Equation (28).

$$\begin{aligned} &Pr(q_i \geq q^*|\vec{V_i}) = \int_{q^*}^{1} Pr(q_i|\vec{V_i})dq_i = \int_{q^*}^{1}\frac{Pr(\vec{V_i}|q_i)}{Pr(\vec{V_i})}Pr(q_i)dq_i \\ &= \frac{\int_{q^*}^{1}\prod_{j=1}^{n} Pr(v_{i_j}|q_i, u_{i_j})dq_i}{\int_{-1}^{1}\prod_{j'=1}^{n} Pr(v_{i_{j'}}|q_i', u_{i_{j'}})dq_i'} \end{aligned} \tag{28}$$

With the knowledge of Equation (23), the $Pr(q_i \geq q^*|\vec{V_i})$ is computed in Equation (29).

$$\begin{aligned} Pr(q_i \geq q^*|\vec{V_i}) &= \frac{\int_{q^*}^{1}\prod_{j=1}^{n} Pr(v_{i_j}|q_i, u_{i_j})dq_i}{\int_{-1}^{1}\prod_{k=1}^{n} Pr(v_{i_k}|q_i', u_{i_k})dq_i'} \\ &= \frac{\sum_{k_1\cdots k_n=1}^{2}\prod_{j=1}^{n}\pi_{k_j}^{i_j}(\mathtt{B}(a_{\mathbf{k}}, b_{\mathbf{k}}) - \mathtt{B}_{H(q^*)}(a_{\mathbf{k}}, b_{\mathbf{k}}))}{\sum_{k_1'\cdots k_n'=1}^{2}\prod_{j'=1}^{n}\pi_{k_{j'}'}^{i_{j'}}\mathtt{B}(a_{\mathbf{k'}}, b_{\mathbf{k'}})} \end{aligned} \tag{29}$$

Formally, given a popularity threshold q^* and a new published item's first n vote $\vec{V_i}$, the ranking score for its popularity is defined as $Pr(q_i \geq q^*|\vec{V_i})$, the probability that this item's popularity q_i is no smaller than a predefined q^*.

[7]Notice that we do not consider the impact of the item's position in the system as in [9].

We again use the example in Section 3 to illustrate this ranking function. The difference lies in the two voters' profiles where $u_1 = \langle 0.9, 0.1 \rangle$ is a Conformer-dominated user and $u_2 = \langle 0.2, 0.8 \rangle$ is a Maverick-dominated person. For $q^* = 0.8$, their popularity ranking scores are derived below.

Given the first item's early vote $\vec{V_1} = \langle (u_1, 1) \rangle$, the ranking score is computed as in Equation (30).

$$
\begin{aligned}
&Pr(q_1 \geq 0.8 | (u_1, 1)) \\
&= \frac{0.9(B(2,1) - B_{0.9}(2,1)) + 0.1(B(1,2) - B_{0.9}(1,2))}{0.9 \times B(2,1) + 0.1 \times B(1,2)} \\
&= 0.172
\end{aligned}
\tag{30}
$$

Similarly, the ranking score of the second item, given the early vote $\vec{V_2} = \langle (u_2, 1) \rangle$, is computed as in Equation (31).

$$
\begin{aligned}
&Pr(q_2 \geq 0.8 | (u_2, 1)) \\
&= \frac{0.2(B(2,1) - B_{0.9}(2,1)) + 0.8(B(1,2) - B_{0.9}(1,2))}{0.2 \times B(2,1) + 0.8 \times B(1,2)} \\
&= 0.046
\end{aligned}
\tag{31}
$$

Therefore the first item is ranked higher than the second one for top-k popular item prediction. This result also agrees with our intuition because the item favored by Conformer will probably be favored by the majority while the one liked by Maverick will usually not receive high attention.

Note that these two ranking functions have different time complexity. Suppose the time cost for computing complete Beta function $\mathtt{B}(a, b)$ is $O(1)$, then the Quality-based one only needs $O(1)$ while the Aggregation-based ranking function requires $O(|U|)$. As for the effectiveness, these two ranking functions show similar performance (as to be shown later in Section 6), suggesting their possible equivalence in the ranking process.

6. EVALUATION

In this section we show the experiment results from the evaluation of our proposed Conformer-Maverick models on top-k popular item prediction. Three algorithms, i.e., user-based collaborative filtering (UCF) [7], singular value decomposition (SVD), and biased SVD (SVD++) [8], are also evaluated for comparison.

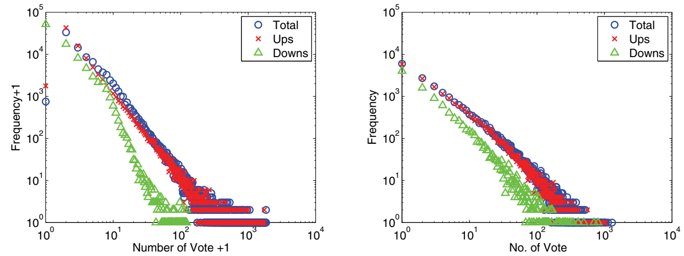
In the following, Section 6.1 introduces the experiment data set as well as the preparation works. Section 6.2 gives the details of our evaluation measures and methodology. Section 6.3 shows the experiment results.

6.1 Data Preparation

The dataset used in our evaluation comes from a popular iPhone application *JokeBox*, a joke sharing platform where people can publish jokes as well as voting others' jokes. Voting an item, a user can only choose "like" or "dislike", corresponding to the positive and negative votes. Different from YouTube and Digg, JokeBox contains detailed information of each user's votes, which can be used to build user profiles for our CM model. The dataset used in this work spans from December 23, 2010 to May 3, 2011.

When preparing the dataset, we only kept items which had received "sufficient votes" and users who had committed "sufficient votes". The sufficient votes, i.e., the threshold σ in Section 2.1, aims to guarantee that i) the item's quality can be well reflected by the votes; and ii) there are enough data instances to learn the user's personality.

Figure 6(a) shows the votes received by an item follow a power-law distribution. As Pareto principle [8] indicates that 20% of members occupy 80% resources in a typical power-law distribution. Therefore, we adopt the minimum number of top 20% items' votes as the threshold (i.e., the "sufficient votes"), which is 8. After removing items whose votes are less than 8, we kept the users who had voted at least one of the remaining jokes. The vote distribution for each user is then shown in Figure 6(b). Again it is a power-law distribution and we set the threshold of votes for an "active" user to be 12 according to the Pareto rule. After this preparation phase, we get 16,490 items, 4,159 user and 353,721 votes.



(a) Item-Vote (b) User-Vote

Figure 6: Vote Distribution

6.2 Evaluation Measures and Methodology

As formulated in Section 2.2, the top-k popular item prediction is a ranking problem. The evaluation is then to evaluate the returned list of k items. In the experiments, we use two widely used ranking measures, i.e., *Root Mean Squared Error (RMSE)* and *Normalized Discount Cumulative Gain (NDCG)*, as defined in Equation (32) and (33).

$$
\mathtt{RMSE}@k = \sqrt{\frac{\sum_{i=1}^{k}(f_i - g_i)^2}{k}} \tag{32}
$$

where the $f_i = i$ and g_i are respectively the predicted and actual rank of the i_{th} item.

$$
\mathtt{NDCG}@k = \frac{\mathtt{DCG}@k}{\mathtt{iDCG}@k} \tag{33}
$$

where DCG is defined as Equation (34) and iDCG is the maximum DCG.

$$
\mathtt{DCG}@k = \begin{cases} G_k & k = 1 \\ \mathtt{DCG}@(k-1) + \frac{G_k}{log_b k} & k > 1 \end{cases} \tag{34}
$$

where $G_k = 1 - \frac{k}{|T|}$ is "Gain" for k_{th} item and $|T|$ is the size of test data set.

Intuitively, for RMSE, a smaller value indicates a better performance while bigger NDCG suggests a better one.

In the experiments, we use collaborative filtering as the baseline solution, which works as follows. It uses the training data to learn either the user similarity (for UCF) or the feature vectors of users and items (for SVD and SVD++). To rank the items in the test data, this method first computes a score, or predicts a rating of each person to these items. The final ranking score is obtained by simply adding all users' predicted ratings.

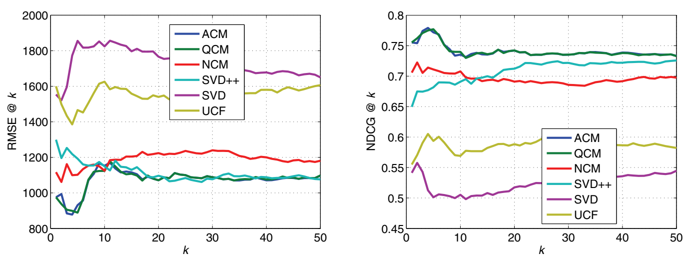
N-Cross-validation is adopted as the evaluation methodology and the average performance is reported.

6.3 Experiment Results

This section discusses the experiment results. In the following, we first show the general performance of tested solutions. Then results on different n (the number of early votes) and different N (training-test data size) are shown.

[8]http://en.wikipedia.org/wiki/Pareto_principle

Figure 7 shows the general performance of the methods on top-k popular item prediction with regarding to different k. In this experiment set, the number of early vote n and the cross-validation group number N are both set to 5. By empirical study, the q^* is set to 0.8 where CM achieves best performance. The CM model with Aggregation-based ranking function is denoted as ACM while the one with Q-based denoted as QCM. As shown in Figure 7(a), the performance of ACM and QCM are nearly undistinguishable, suggesting the equivalence of the two ranking functions. However, the Q-based ranking function is preferred considering its higher efficiency. In the rest of the experiments, the Q-based ranking function is used for representing methods based on our CM model (and thus denoted as CM). Also from the figure we can see that when $k \leq 10$, our proposed CM model displays the best performance in terms of RMSE. After $k \geq 10$, the performance of CM model and SVD++ is hard to distinguish. However, for NDCG, the performance of the CM model is always better than the other solutions with regarding to all k, as shown in Figure 7(b). For the naive CM model, its performance is the second best when $k \leq 10$ but is worse than SVD++ for a bigger k. From this general performance comparison, we can see that our (naive) CM model does a good job in popularity prediction for smaller k. Although our models are no better than SVD++ for big k (i.e., $k \geq 10$), it is very valuable because users, especially mobile phone users, may only be able to consume items in a short top-ranking list instead of the whole long list of recommendation. We also argue that the failure of SVD++ may come from the cold start issue since the ranking is based on only the first 5 votes on a new item.

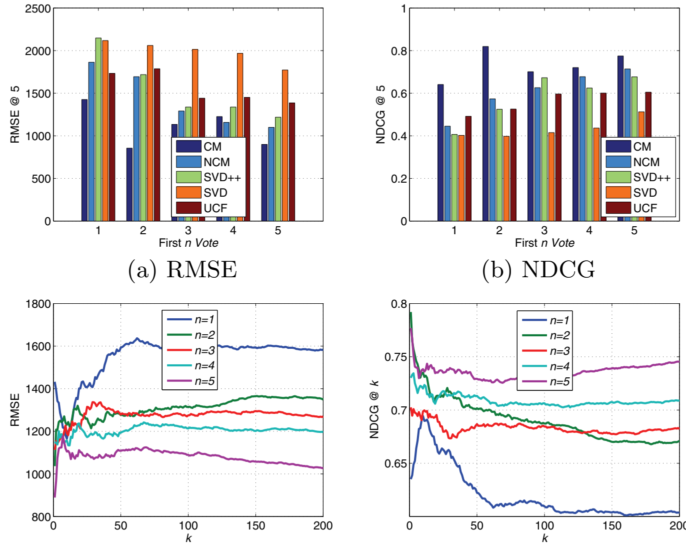


(a) RMSE (b) NDCG

Figure 7: General Comparison

The second experiment aims to show the impact of the number of early votes n on the performance. In this experiment set, we varies n from 1 to 5 and show the top-5 prediction performance in Figure 8. For both RMSE and NDCG, our proposed CM and NCM model outperforms the baseline solutions. More specifically, the performance of all solutions except for CM, degrade as n decreases. This shows the strength of our CM in making good prediction with few early votes. For NCM and collaborative filtering solutions, when more early votes are involved, more information can be relied on and thus the prediction will become more accurate, resulting in the performance improvement. Notice that it is not clear to us whether the performance of the CM model only prevail when the top-5 list is considered. To further demonstrate the superior performance of our CM model, we plot Figure 8(c) and 8(d) to display its performance under different n and k. We can see that its improvement is clear along the increase of n when k is bigger.

For the third set of experiments, we show the impact of training-test data size, i.e., N. A larger N means larger amount of training data and smaller quantity of test data and vice versa. The results are shown in Figure 9. Generally, the performance of all solutions shows some improvement

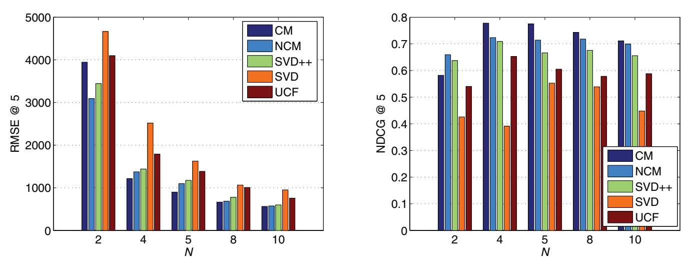


(a) RMSE (b) NDCG

(c) RMSE for CM (d) NDCG for CM

Figure 8: Evaluation on Different n

with larger N since there are more training data. The NCM outperforms the baseline solutions in all cases. On the other hand, the CM shows the best performance except for $N = 2$, where only 50% of data is available for training. In other words, the sparsity of the training data has an impact on the CM model, which suffers inaccuracy when modeling the user's CM personality with insufficient training data.



(a) RMSE (b) NDCG

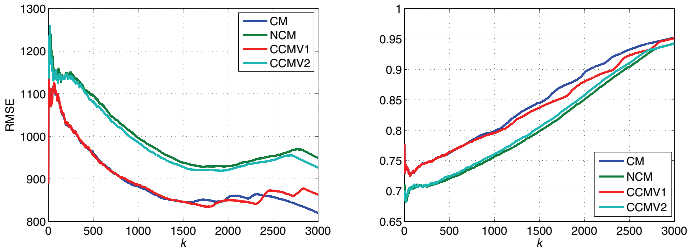
Figure 9: Evaluation on Different N

6.4 Comparison of CM and NCM Models

As shown in previous results, the two proposed models, i.e., CM and NCM, display different performance. In this section, we made a detailed comparison of their differences.

Although their basic ideas are similar, CM adopts a more sophisticated process to model user's voting behavior while NCM is rather heuristic and straightforward. Specifically, the differences are attribute to two aspects: i) the modeling of user's personality, i.e., the learning of π_c, π_m for CM and the calculation of w_c, w_m for NCM; ii) the ranking function. Since the personality parameters have the same value range ($\pi_c, \pi_m, w_c, w_m \in [0, 1]$) and constraint ($\pi_c + \pi_m = 1, w_c + w_m = 1$), two new solutions can be "generated" by combing users' personality and ranking function from either model.

In the experiment we evaluate the performance of two new solutions: 1) *Combined CM Variance 1 (CCMV1)*, adopting the personality model of CM and the ranking function of NCM; and 2) *Combined CM Variance 2 (CCMV2)*, combing the personality model of NCM and the ranking function of CM. It can be seen from Figure 10 that i) the performances of CM and CCMV1 are significantly better than those of CCMV2 and NCM; ii) the performance of CM and CCMV1 are similar while the performance of CCMV2 is slightly better than that of NCM. Therefore we can safely conclude that the model of user personality is the major factor that affects the performance difference of CM and NCM.



(a) RMSE (b) NDCG

Figure 10: Combined CM model

Recall in Section 4.4 we discuss the difference of NCM and CM in modeling user profiles. CM is more robust in handling the observed vote data while NCM is more "radical", attributing each vote to either Conformer weight w_c or Maverick weight w_m. The result here supports our claim that the CM model can better fit the data and outperforms NCM, although the two methods share the same basic ideas.

7. RELATED WORK

With the growing popularity of content-sharing services in web and mobile phones, research on the online contents' popularity evolution [3, 5, 17] and prediction are growing.

Szabo and Huberman [13] studied the change of votes for items in both Digg.com and YouTube. They analyzed the popularity evolution and found that the growth ratio is $(1 + rX)$, where r is a time-sensitive parameter and X is a variable that satisfies a normal distribution. This method aims to explore the popularity growing trend and can not rank the popularity of a single item. Also, prediction suffers inaccuracy if it is purely based on the number of early votes.

In [18], Yu et. al. exploited the content to judge whether a marketing message would be popular. Two classification methods, i.e., SVM and Naive Bayes are used and the feature is modeled as a vector of words. Their methods, however, are limited only to textual contents. Furthermore, the classification tools are unable to process the ranking problems addressed in this paper.

Lerman and Hogg [9] proposed a stochastic model that considers both social influence and the layout of the website. In this model, the increasing rate of the votes is related to the possibility that a user finds this item i) in font page, ii) in upcoming page and iii) through her friends. Prediction can then be made after learning the three parameters from the training data. This method, however, can not work in our scenario. Firstly, we are interested in the potential for newly published items to be popular instead of the position difference. Also, there is no social network in our data set and the voting is free of social influence. Finally, we focus on the voting process of a person, a key behavior that determines the popularity of an item.

Other existing works integrate user comments [12, 15, 16, 14]. This information, however, is not always available compared with votes. For example, a person is more likely to vote than leave a comment as the former one takes much less time and effort.

To improve the prediction precision, some works resort to information of other social media, e.g., web blog [6, 11], Twitter [1, 2]. While this is an interesting direction, this method works only when there are multiple sources containing the information of items, such as book sales [6], movie revenues [11, 1, 2] and stock markets [2]. Also, mining external web sites introduces high cost compared to methods that only rely on the information within the system.

8. CONCLUSION AND FUTURE WORK

In this work we propose a *Conformer-Maverick (CM)* model to rank potentially popular items. The CM assumes each person has two personalities, i.e., Conformer and Maverick, which guide the voting behavior. Conformers' votes are usually consistent with the majority's opinions while Mavericks' votes conflicting. Different people have different distributions of these two personalities, which can be learned from the voting history. By exploiting such behaviors, potentially popular items can be ranked by their early voters.

While people's topic distributions tend to be stable for items without many complex genres, e.g., jokes, they may vary when it comes to items with multiple genres, e.g., action, romance, horror and so on in movies. In our future work, we plan to adapt our CM model for such complex items and applications.

9. REFERENCES

[1] S. Asur and B. A. Huberman. Predicting the future with social media. *WIIAT*, 1:492–499, 2010.

[2] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[3] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Trans. Netw.*, 17:1357–1370, 2009.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[5] F. Figueiredo, F. Benevenuto, and J. M. Almeida. The tube over time: characterizing popularity growth of youtube videos. In *WSDM*, pages 745–754, 2011.

[6] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *KDD*, pages 78–87, 2005.

[7] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, pages 230–237, 1999.

[8] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

[9] K. Lerman and T. Hogg. Using a model of social dynamics to predict popularity of news. In *WWW*, pages 621–630, 2010.

[10] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 2007.

[11] G. Mishne. Predicting movie sales from blogger sentiment. In *AAAI-CAAW*, 2006.

[12] G. Mishne and N. Glance. Leave a reply: An analysis of weblog comments. In *In Third annual workshop on the Weblogging ecosystem*, 2006.

[13] G. Szabo and B. o. A. Huberman. Predicting the popularity of online content. *Communications of the Acm*, 53(8):80–88, 2010.

[14] A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida. Predicting the popularity of online articles based on user comments. In *WIMS*, pages 67:1–67:8, 2011.

[15] M. Tsagkias, M. Larson, and M. d. Rijke. Exploiting surface features for the prediction of podcast preference. *Advances in Information Retrieval*, 5478:473–484, 2009.

[16] M. Tsagkias, M. Larson, and M. d. Rijke. News comments:exploring, modeling, and online prediction. *Advances in Information Retrieval*, 5993:191–203, 2010.

[17] D. M. Wilkinson. Strong regularities in online peer production. In *EC*, pages 302–309, 2008.

[18] B. Yu, M. Chen, and L. Kwok. Toward predicting popularity of social marketing messages. *Computer Science*, 6589:317–324, 2011.